Types of Parallel Computers

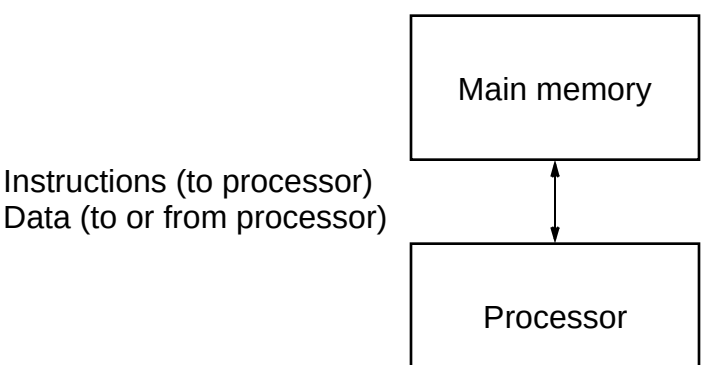
Two principal types:

- Shared memory multiprocessor
- Distributed memory multicomputer

Shared Memory Multiprocessor

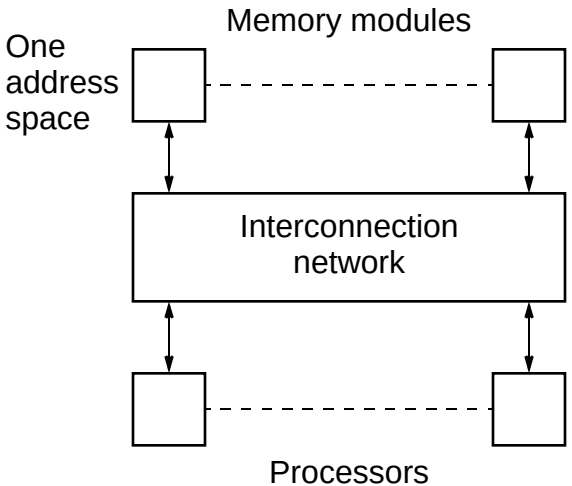
Conventional Computer

Consists of a processor executing a program stored in a (main) memory:

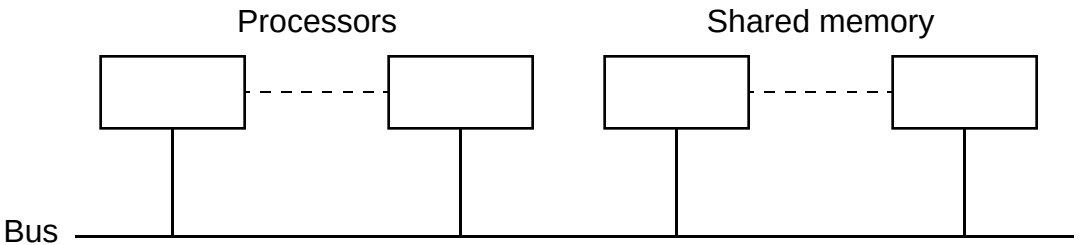


Each main memory location located by its *address*. Addresses start at 0 and extend to $2^b - 1$ when there are b bits (binary digits) in address.

Natural way to extend single processor model - have multiple processors connected to multiple memory modules, such that each processor can access any memory module - so-called *shared memory* configuration:



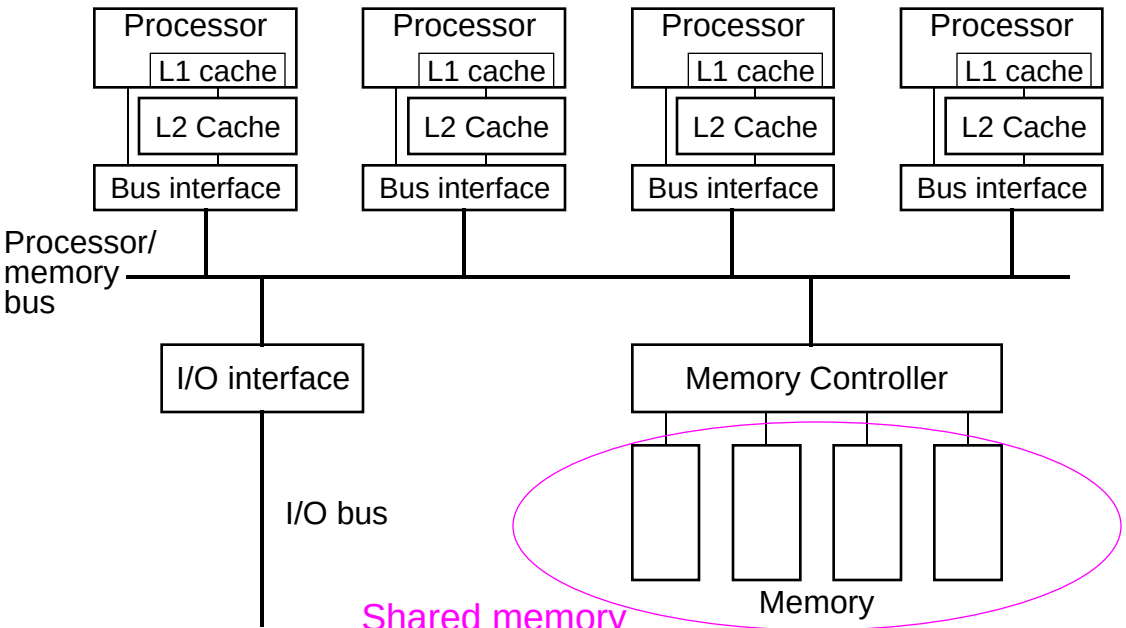
Simplistic view of a small shared memory multiprocessor



Examples:

- Dual Pentiums
- Quad Pentiums

Quad Pentium Shared Memory Multiprocessor

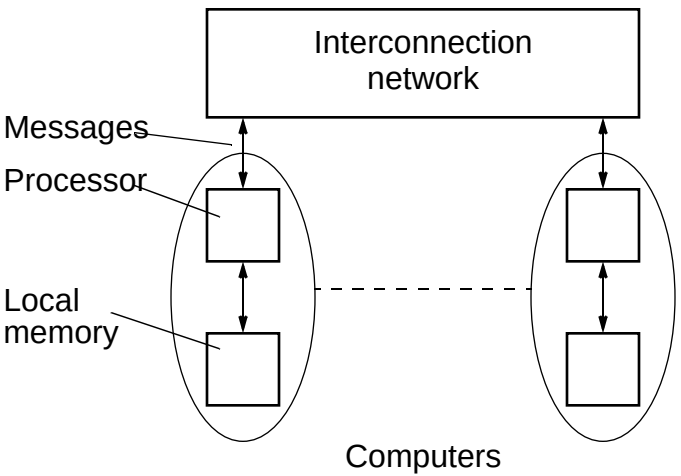


Programming Shared Memory Multiprocessors

- Threads - programmer decomposes the program into individual parallel sequences, (threads), each being able to access variables declared outside threads.
  Example Pthreads
- A sequential programming language with preprocessor compiler directives to declare shared variables and specify parallelism.
  Example OpenMP - industry standard - needs OpenMP compiler
- A sequential programming language with added syntax to declare shared variables and specify parallelism.
  Example UPC (Unified Parallel C) - needs a UPC compiler.
- A parallel programming language with syntax to express parallelism, in which the compiler creates the appropriate executable code for each processor (not now common)
- A sequential programming language and ask a parallelizing compiler to convert it into parallel executable code. - also not now common

Message-Passing Multicomputer

Complete computers connected through an interconnection network:



Interconnection Networks

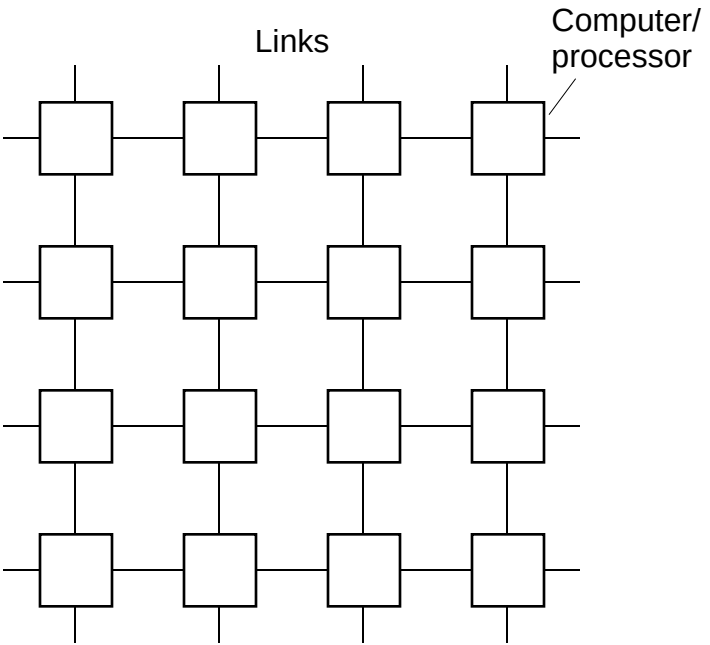
With direct links between computers

- Exhausive connections
- 2-dimensional and 3-dimensional meshs
- Hypercube

Using Switches:

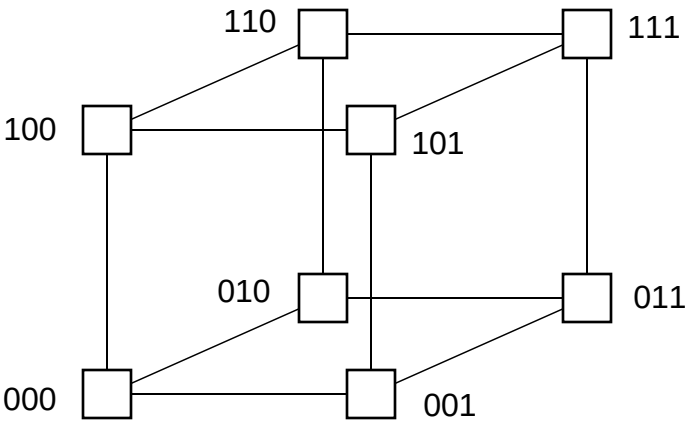
- Crossbar
- Trees
- Multistage interconnection networks

Two-dimensional array (mesh)

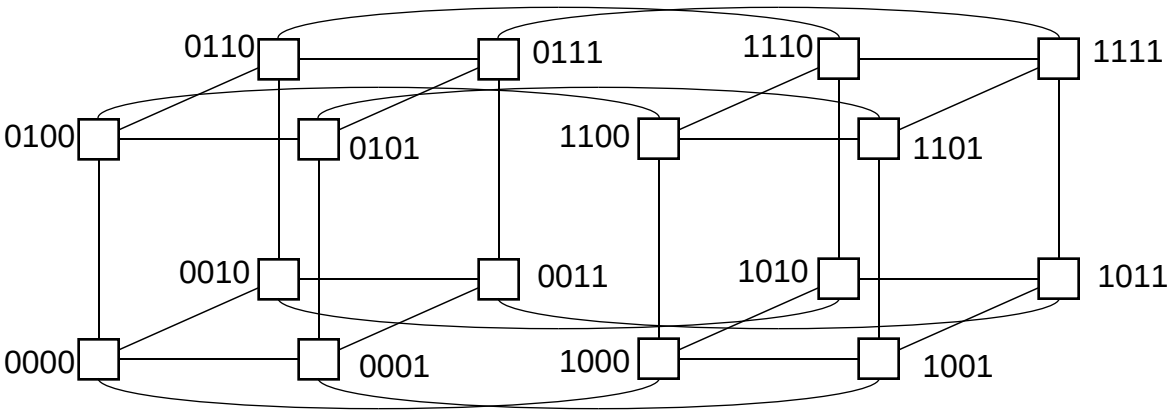


Also three-dimensional - used in some large high performance systems.

Three-dimensional hypercube

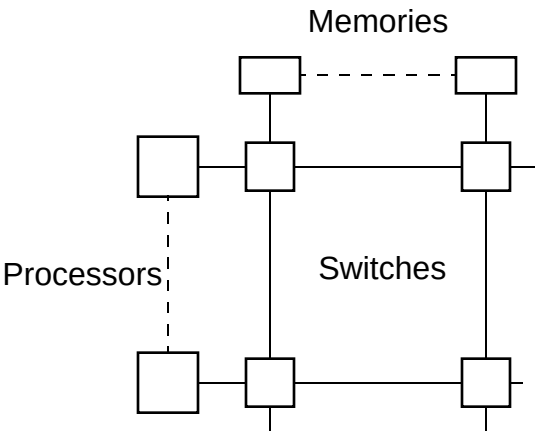


Four-dimensional hypercube

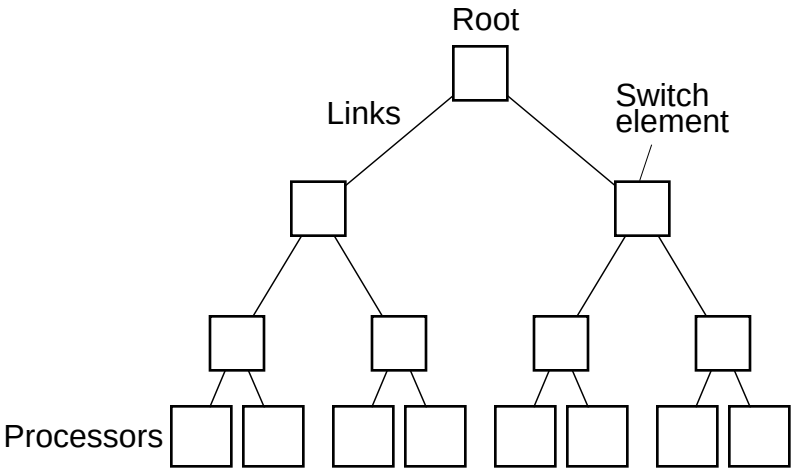


Hypercubes popular in 1980's - not now

Crossbar switch

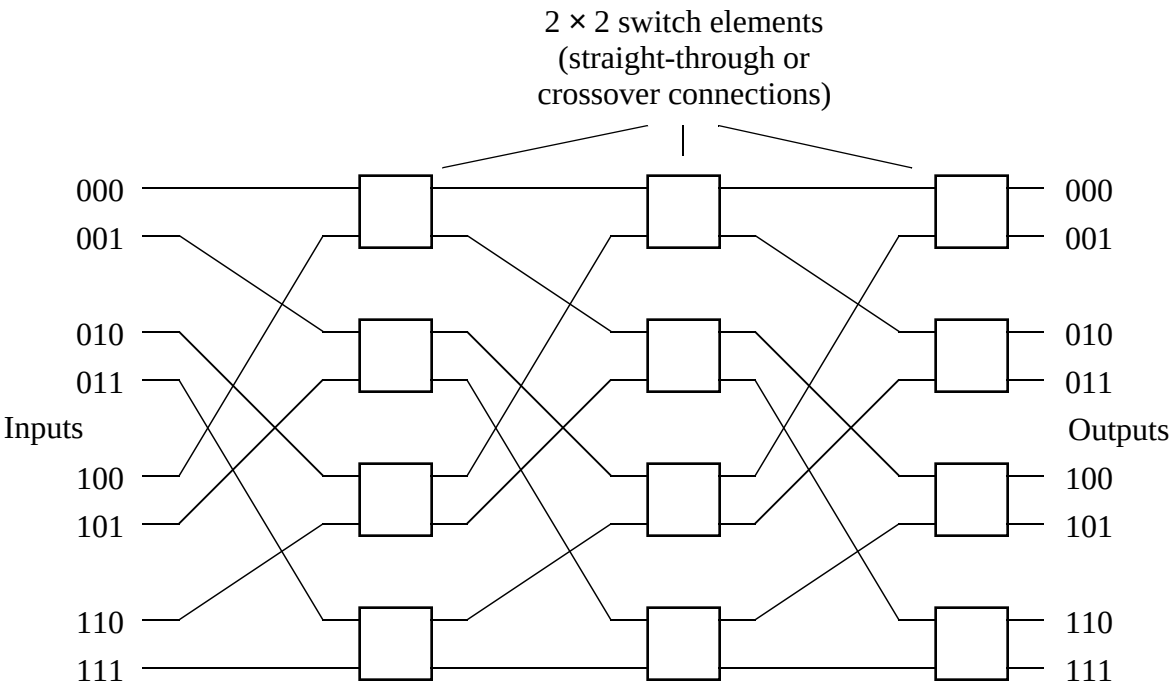


Tree



Multistage Interconnection Network

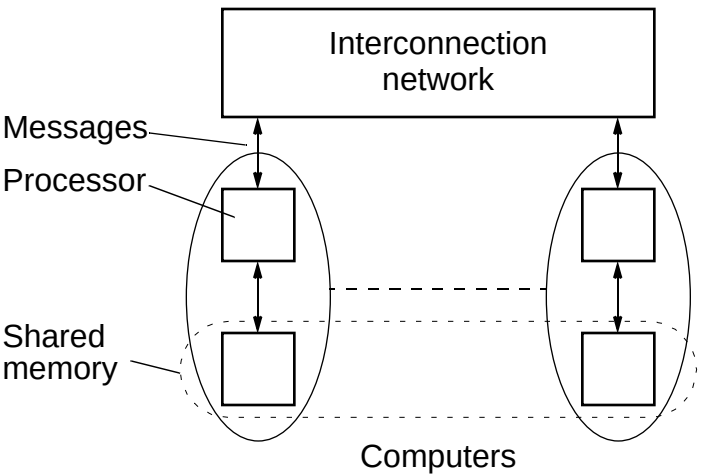
Example: Omega network



Distributed Shared Memory

Making the main memory of a group of interconnected computers look as though it is a single memory with a single address space.

Then can use shared memory programming techniques.



Flynn's Classifications

Flynn (1966) created a classification for computers based upon instruction streams and data streams:

Single instruction stream-single data stream (SISD) computer

In a single processor computer, a single stream of instructions is generated from the program. The instructions operate upon a single stream of data items. Flynn called this single processor computer a s*ingle instruction stream-single data stream* (SISD) computer.

Multiple Instruction Stream-Multiple Data Stream (MIMD) Computer

General-purpose multiprocessor system - each processor has a separate program and one instruction stream is generated from each program for each processor. Each instruction operates upon different data.

Both the shared memory and the message-passing multiprocessors so far described are in the MIMD classification.

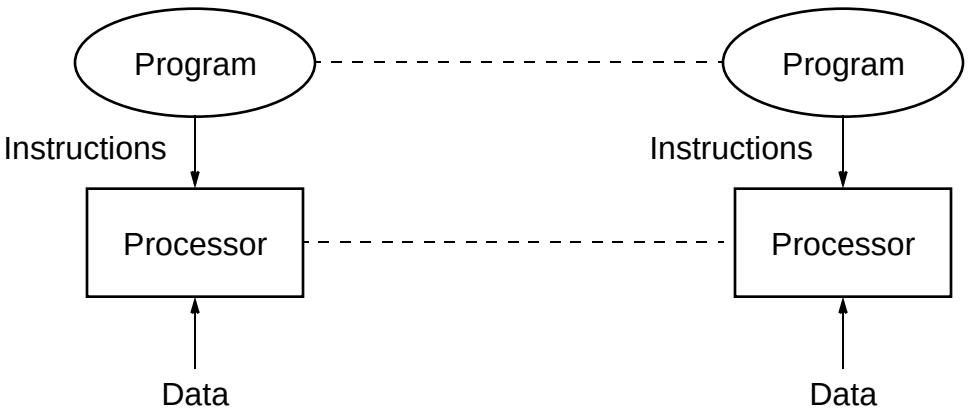
Single Instruction Stream-Multiple Data Stream (SIMD) Computer

A specially designed computer in which a single instruction stream is from a single program, but multiple data streams exist. The instructions from the program are broadcast to more than one processor. Each processor executes the same instruction in synchronism, but using different data.

Developed because there are a number of important applications that mostly operate upon arrays of data.

Multiple Program Multiple Data (MPMD) Structure

Within the MIMD classification, which we are concerned with, each processor will have its own program to execute:



Single Program Multiple Data (SPMD) Structure

Single source program is written and each processor will execute its personal copy of this program, although independently and not in synchronism.

The source program can be constructed so that parts of the program are executed by certain computers and not others depending upon the identity of the computer.